Listing of Claims

1.   (Canceled)

2.   (Previously Presented) A method for updating data in a storage device, comprising:

receiving an update to one or more blocks of customer data at addresses in the storage device;

for each block of data to update, generating metadata indicating the address of the block in the storage device and an error checking code that is capable of being used to determine whether the customer data in the block has changed;

for each block of data to update, writing the block of data to update and the metadata for the block to cache;

for each block of data to update, determining whether the address of the block of data in the metadata and the address in the storage device to update match;

for each block of data to update, performing an operation on the customer data in the block and the error checking code to determine whether the customer data has changed; and

transferring the block of data to update and metadata for the block to the storage device if the address of the block in the metadata and requested address match and the customer data has not changed.

3.   (Original) The method of claim 2, wherein the operation performed on the customer data comprises XORing the customer data and wherein the error checking code comprises a longitudinal redundancy checking code.

4.   (Currently Amended) A method for updating data in a storage device, comprising:

receiving an update to one or more blocks of customer data at addresses in the storage device;

setting up a control block including the address of ~~a first~~ the block of data to update in the storage device and an instruction to generate the address and error code as metadata for the block;

BEST AVAILABLE COPY

for each block of data to update, generating metadata indicating the address in the control block as the address of the block in the storage device to write and an error checking code that is capable of being used to determine whether the customer data in the block has changed;

for each block of data to update, writing the block of data to update and the metadata for the block to cache; and

for each block of data to update, transferring the block of data and the metadata for the block from the cache to the storage device.

5.   (Original) The method of claim 4, further comprising:

for each block to update, incrementing the block address in the control block before processing a next block, wherein the incremented block address is used as the address in the storage device to generate as metadata for the next block of data to update.

6.   (Previously Presented) A method for updating data in a storage device, comprising:

receiving an update to one or more blocks of customer data at addresses in the storage device;

for each block of data to update, generating metadata indicating the address of the block in the storage device and an error checking code, wherein the error checking code is generated by XORing the customer data in the block such that the error checking code comprises a longitudinal redundancy checking code, and wherein the error checking code is used to determine whether the customer data in the block has changed;

for each block of data to update, writing the block of data to update and the metadata for the block to cache; and

for each block of data to update, transferring the block of data and the metadata for the block from the cache to the storage device.

7.   (Previously Presented) The method of claim 2, wherein generating the metadata and determining whether the address of the block in the storage device and block address in metadata match and performing the operation on the customer data and error checking is

Page 3 of 17

performed by a device that is separate from a main processor, wherein the device transfers the block of data from the cache to the storage device using a direct memory access (DMA) channel.

8.   (Currently Amended) The method of claim 2, further comprising:
setting up a control block in the cache including the block address of a [[first]] the block to update and an instruction to check the address and error code of the block, wherein determining whether the block address in the metadata in the storage device and the block address to update match comprises using the block address in the control block in the cache as the block address to update to compare with the block address in the metadata stored with the block in the cache.

9.   (Currently Amended) The method of claim 8, further comprising:
for each requested block, incrementing the block address in the first control block after transferring the block from the cache to the storage device, wherein the incremented block address is used as the block address to compare with the block address in the metadata in the cache for the next requested block.

10.   (Previously Presented) A method for updating data in a storage device, comprising:
receiving an update to one or more blocks of customer data at addresses in the storage device;
for each block of data to update, generating metadata indicating the address of the block in the storage device and an error checking code that is capable of being used to determine whether the customer data in the block has changed;
for each block of data to update, writing the block of data to update and the metadata for the block to cache;
for each block of data to update, transferring the block of data and the metadata for the block from the cache to the storage device;
recovering from a power loss; and

using the metadata for blocks in cache to rebuild cache control blocks for the blocks in cache after recovering from the power loss.

11.   (Original) The method of claim 10, wherein the blocks of data and metadata are stored in a non-volatile portion of the cache and the cache control blocks are stored in a volatile portion of the cache.

12.   (Original) The method of claim 10, further comprising:

maintaining a first data structure indicating whether each block of data in cache is valid or invalid and a second data structure indicating whether each block of data includes modified or unmodified data;

for each block of data in the cache, using the error checking code to determine whether the block of data in the cache has changed; and

for each block of data in the cache, indicating in the first data structure that the block of data is invalid if the second data structure indicates that the block of data is not modified and the block of data has changed.

13.   (Original) The method of claim 10, further comprising:

maintaining a first data structure indicating whether each block of data in cache is valid or invalid and a second data structure indicating whether each block of data includes modified or unmodified data;

for each block of data in the cache, using the error checking code to determine whether the block of data in the cache has changed; and

for each block of data in the cache accessing the update to the block from another storage location if the second data structure indicates that the block of data is modified and the block of data has changed.

14.   (Original) The method of claim 10, further comprising:

maintaining a first data structure indicating whether each block of data in cache is valid or invalid and a second data structure indicating whether each block of data includes modified or unmodified data;

for each block of data in the cache, using the error checking code to determine whether the block of data in the cache has changed; and

rebuilding the cache control block for the block using the address information in the metadata for the block if the block of data has not changed.

15.   (Currently Amended) The method of claim [[1]] 4, wherein the error checking code is further capable of being used to determine whether the metadata in the block has changed.

16.   (Canceled)

17.   (Previously Presented) A system for updating data, comprising:

a storage device;

a cache;

means for receiving an update to one or more blocks of customer data at addresses in the storage device;

means for generating metadata, for each block of data to update, indicating the address of the block in the storage device and an error checking code that is capable of being used to determine whether the customer data in the block has changed;

means for writing the block of data to update and the metadata for the block to the cache for each block of data to update;

means for determining whether the address of the block of data in the metadata and the address in the storage device to update match for each block of data to update;

means for performing an operation on the customer data in the block and the error checking code to determine whether the customer data has changed for each block of data to update; and

Page 6 of 17

means for transferring the block of data to update and metadata for the block to the storage device if the address of the block in the metadata and requested address match and the customer data has not changed.

18.   (Original) The system of claim 17, wherein the operation performed on the customer data comprises XORing the customer data and wherein the error checking code comprises a longitudinal redundancy checking code.

19.   (Currently Amended) A system for updating data, comprising:

a storage device;

a cache;

means for receiving an update to one or more blocks of customer data at addresses in the storage device;

means for setting up a control block including the address of a first the block of data to update in the storage device and an instruction to generate the address and error code as metadata for the block;

means for generating metadata, for each block of data to update, indicating the address in the control block as the address of the block in the storage device to write and an error checking code that is capable of being used to determine whether the customer data in the block has changed;

means for writing the block of data to update and the metadata for the block to the cache for each block of data to update; and

means transferring the block of data and the metadata for the block from the cache to the storage device for each block of data to update.

20.   (Original) The system of claim 19, further comprising:

means for incrementing the block address in the control block, for each block to update, before processing a next block, wherein the incremented block address is used as the address in the storage device to generate as metadata for the next block of data to update.

Page 7 of 17

21.   (Previously Presented) A system for updating data, comprising:

a storage device;

a cache;

means for receiving an update to one or more blocks of customer data at addresses in the storage device;

means for generating metadata, for each block of data to update, indicating the address of the block in the storage device and an error checking code, wherein the error checking code is generated by XORing the customer data in the block such that the error checking code comprises a longitudinal redundancy checking code, wherein the error checking code is used to determine whether the customer data in the block has changed;

means for writing the block of data to update and the metadata for the block to the cache for each block of data to update; and

means transferring the block of data and the metadata for the block from the cache to the storage device for each block of data to update.

22.   (Original) The system of claim 17, wherein the means for generating the metadata, determining whether the address of the block in the storage device and block address in metadata match, and performing the operation on the customer data and error checking comprises a device that is separate from a main processor, wherein the device transfers the block of data from the cache to the storage device using a direct memory access (DMA) channel.

23.   (Currently Amended) The system of claim 17, further comprising:

means for setting up a control block in the cache including the block address of ~~a first~~ the block to update and an instruction to check the address and error code of the block, wherein the means for determining whether the block address in the metadata in the storage device and the block address to update match comprises using the block address in the control block in the cache as the block address to update to compare with the block address in the metadata stored with the block in the cache.

Page 8 of 17

24. (Currently Amended) The system of claim 23, further comprising:

means for incrementing the block address in the [[first]] control block, for each requested block, after transferring the block from the cache to the storage device, wherein the incremented block address is used as the block address to compare with the block address in the metadata in the cache for the next requested block.

25. (Previously Presented) A system for updating data, comprising:

a storage device;

a cache;

means for receiving an update to one or more blocks of customer data at addresses in the storage device;

means for generating metadata, for each block of data to update, indicating the address of the block in the storage device and an error checking code that is capable of being used to determine whether the customer data in the block has changed;

means for writing the block of data to update and the metadata for the block to the cache for each block of data to update;

means transferring the block of data and the metadata for the block from the cache to the storage device for each block of data to update;

means for recovering from a power loss; and

means for using the metadata for blocks in cache to rebuild cache control blocks for the blocks in cache after recovering from the power loss.

26. (Previously Presented) The system of claim 25, wherein the blocks of data and metadata are stored in a non-volatile portion of the cache and the cache control blocks are stored in a volatile portion of the cache.

27. (Previously Presented) The system of claim 25, further comprising:

means for maintaining a first data structure indicating whether each block of data in cache is valid or invalid and a second data structure indicating whether each block of data includes modified or unmodified data;

means for using the error checking code, for each block of data in the cache, to determine whether the block of data in the cache has changed; and

means for indicating in the first data structure, for each block of data in the cache, that the block of data is invalid if the second data structure indicates that the block of data is not modified and the block of data has changed.

28.   (Previously Presented) The system of claim 25, further comprising:

means for maintaining a first data structure indicating whether each block of data in cache is valid or invalid and a second data structure indicating whether each block of data includes modified or unmodified data;

means for using the error checking code, for each block of data in the cache, to determine whether the block of data in the cache has changed; and

means for accessing, for each block of data in the cache, the update to the block from another storage location if the second data structure indicates that the block of data is modified and the block of data has changed.

29.   (Previously Presented) The system of claim 25, further comprising:

means for maintaining a first data structure indicating whether each block of data in cache is valid or invalid and a second data structure indicating whether each block of data includes modified or unmodified data;

means for using the error checking code to determine whether the block of data in the cache has changed for each block of data in the cache; and

means for rebuilding the cache control block for the block using the address information in the metadata for the block if the block of data has not changed.

30.   (Currently Amended) The system of claim [[16]] 19, further comprising means for using the error checking code to determine whether the metadata in the block has changed.

31.   (Canceled)

32.   (Previously Presented) An integrated circuit device including logic for updating data in a storage device, wherein the integrated circuit device is in data communication with a cache, wherein the logic performs:

receiving an update to one or more blocks of customer data at addresses in the storage device;

for each block of data to update, generating metadata indicating the address of the block in the storage device and an error checking code that is capable of being used to determine whether the customer data in the block has changed;

for each block of data to update, writing the block of data to update and the metadata for the block to the cache;

for each block of data to update, determining whether the address of the block of data in the metadata and the address in the storage device to update match;

for each block of data to update, performing an operation on the customer data in the block and the error checking code to determine whether the customer data has changed; and

transferring the block of data to update and metadata for the block to the storage device if the address of the block in the metadata and requested address match and the customer data has not changed.

33.   (Previously Presented) The integrated circuit device of claim 32, wherein the operation performed on the customer data comprises XORing the customer data and wherein the error checking code comprises a longitudinal redundancy checking code.

34.   (Currently Amended) An integrated circuit device including logic for updating data in a storage device, wherein the integrated circuit device is in data communication with a cache and processor, wherein the logic performs:

receiving an update to one or more blocks of customer data at addresses in the storage device;

accessing a control block, generated by the processor, including the address of a first the block of data to update in the storage device and an instruction to cause the integrated circuit device to generate the address and error code as metadata for the block;

Page 11 of 17

for each block of data to update, generating the metadata indicating the address in the control block as the address of the block in the storage device to write and an error checking code that is capable of being used to determine whether the customer data in the block has changed;

for each block of data to update, writing the block of data to update and the metadata for the block to the cache; and

for each block of data to update, transferring the block of data and the metadata for the block from the cache to the storage device.

35. (Previously Presented) The integrated circuit device of claim 34, wherein the logic further performs:

for each block to update, incrementing the block address in the control block before processing a next block, wherein the incremented block address is used as the address in the storage device to generate as metadata for the next block of data to update.

36. (Previously Presented) An integrated circuit device including logic for updating data in a storage device, wherein the integrated circuit device is in data communication with a cache, wherein the logic performs:

receiving an update to one or more blocks of customer data at addresses in the storage device;

for each block of data to update, generating metadata indicating the address of the block in the storage device and an error checking code, wherein the error checking code is generated by XORing the customer data in the block such that the error checking code comprises a longitudinal redundancy checking code, wherein the error checking code is used to determine whether the customer data in the block has changed;

for each block of data to update, writing the block of data to update and the metadata for the block to the cache; and

for each block of data to update, transferring the block of data and the metadata for the block from the cache to the storage device.

Page 12 of 17

37.   (Previously Presented) The integrated circuit device of claim 32, wherein the integrated circuit device is included in a storage system having a main processor, wherein the device transfers the block of data from the cache to the storage device using a direct memory access (DMA) channel.

38.   (Currently Amended) The integrated circuit device of claim 32, wherein a processor generates a control block in the cache including the block address of ~~a first~~ the block to update and an instruction to cause the integrated circuit device to check the address and error code of the block, wherein the integrated circuit device determines whether the block address in the metadata in the storage device and the block address to update match by using the block address in the control block in the cache as the block address to update to compare with the block address in the metadata stored with the block in the cache.

39.   (Currently Amended) The integrated circuit device of claim 38, wherein the logic further performs:

for each requested block, incrementing the block address in the [[first]] control block after transferring the block from the cache to the storage device, wherein the incremented block address is used as the block address to compare with the block address in the metadata in the cache for the next requested block.

40.   (Previously Presented) An integrated circuit device including logic for updating data in a storage device, wherein the integrated circuit device is in data communication with a cache, wherein the logic performs:

receiving an update to one or more blocks of customer data at addresses in the storage device;

for each block of data to update, generating metadata indicating the address of the block in the storage device and an error checking code that is capable of being used to determine whether the customer data in the block has changed;

for each block of data to update, writing the block of data to update and the metadata for the block to the cache;

for each block of data to update, transferring the block of data and the metadata for the block from the cache to the storage device;

recovering from a power loss; and

using the metadata for blocks in cache to rebuild cache control blocks for the blocks in cache after recovering from the power loss.

41.   (Previously Presented) The integrated circuit device of claim 40, wherein the blocks of data and metadata are stored in a non-volatile portion of the cache and the cache control blocks are stored in a volatile portion of the cache.

42.   (Currently Amended) The integrated circuit device of claim [[31]] 34, wherein the error checking code is further capable of being used to determine whether the metadata in the block has changed.

43.   (Canceled)

44.   (Currently Amended) A computer readable medium including at least one data structure used for updating data in a storage device and a cache, comprising:

blocks of customer data;

a block of metadata for each block of customer data, wherein the metadata includes the address of the block in the storage device and an error checking code that is capable of being used to determine whether the customer data in the block has changed while the block is in the cache, wherein a block of data to update and the metadata for the block are written to cache; and

a control block including the address of a first the block of data to update in the storage device and an instruction to generate the address and error code as metadata for the block, wherein generating the metadata indicating the address of the block in the storage device comprises using the block address in the control block as the address of the block in the storage device to write as the metadata.

Page 14 of 17

45.   (Previously Presented) The computer readable medium of claim 44, wherein the block address in the control block is incremented before processing a next block, wherein the incremented block address is used as the address in the storage device to generate as metadata for the next block of data to update.

46.   (Currently Amended) ~~The computer readable medium of claim 43, further comprising:~~ A computer readable medium including at least one data structure used for updating data in a storage device and a cache, comprising:

blocks of customer data;

a control block in the cache including the block address of a [[first]] block to update and an instruction to check the address and error code of the block[[,]]; and

a block of metadata for each block of customer data, wherein the metadata includes the address of the block in the storage device and an error checking code, wherein the error checking code is generated from the customer data and used to determine whether the customer data in the block has changed, wherein ~~determining whether~~ the block address in the metadata in the storage device and the block address to update are determined to match ~~comprises~~ by using the block address in the control block in the cache as the block address to update to compare with the block address in the metadata stored with the block in the cache wherein the block of data to update and the metadata for the block are written to cache; and

47.   (Previously Presented) The computer readable medium of claim 46, wherein the blocks of data and metadata are stored in a non-volatile portion of the cache and the cache control block is stored in a volatile portion of the cache.

48.   (Currently Amended) ~~The computer readable medium of claim 43, further comprising:~~ A computer readable medium including at least one data structure used for updating data in a storage device and a cache, comprising:

blocks of customer data;

a block of metadata for each block of customer data, wherein the metadata includes the address of the block in the storage device and an error checking code, wherein the error checking

code is generated from the customer data and that is used to determine whether the customer data in the block has changed, wherein block of data to update and the metadata for the block are written to cache;

a first data structure indicating whether each block of data in cache is valid or invalid; and

a second data structure indicating whether each block of data includes modified or unmodified data, wherein the error checking code is used to determine whether each block of data in the cache has changed in the event of a data recovery event, wherein the first data structure is modified to indicate that a block of data is invalid if the second data structure indicates that the block of data is not modified and the block of data has changed.

49-51. (Canceled)

Page 16 of 17